

JOHNSON GRANT
IN-62-CR
15136

P.17

COOPERATIVE PROCESSING USER INTERFACES FOR AdaNET

(NASA-CR-188239) COOPERATIVE PROCESSING
USER INTERFACES FOR AdaNET (Softech) 17 p
CSCL 09B

N91-25686

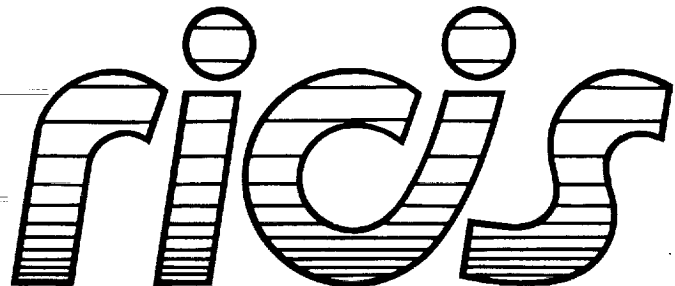
Unclas
G3/62 0015136

Kurt M. Gutzmann

April 30, 1991

Cooperative Agreement NCC 9-16
Research Activity No. SE.18
Deliverable 2.2.1

NASA Johnson Space Center
Information Systems Directorate
Information Technology Division



Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T · E · C · H · N · I · C · A · L R · E · P · O · R · T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

***COOPERATIVE PROCESSING
USER INTERFACES FOR AdaNET***

Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by SofTech, Incorporated. Dr. Charles McKay served as RICIS research representative.

Funding has been provided by Information Technology Division, Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Ernest M. Fridge, of the Software Technology Branch, Information Technology Division, Information Systems Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.



**Cooperative Processing
User Interfaces for AdaNET**
ADANET-FD-R&T-085-0

Kurt M. Gutzmann

April 30, 1991

Subcontract No. 044
Cooperative Agreement NCC9-16
Project No. RICIS No. SE.18

Submitted to:
MountainNet, Inc.
P.O. Box 370
Dellslow, WV 26531-0370

Prepared by:
SofTech, Inc.
1300 Hercules Drive
Suite 105
Houston, TX 77058

Introduction

This report examines cooperative processing user interface (CUI) technology and its possible application to AdaNET. A cooperative processing user interface system shares the task of graphical display generation and presentation between the user's computer and a remote host. The communications link between the two computers is separate matter, but is typically a modem or Ethernet. An example of a cooperative processing user interface is the X-windows system [X11], although the X-client still transmits graphics for display on the server. The two main purposes of a CUI are:

- reduction of the amount of data transmitted between user and host machines¹
- provision of a graphical user interface system to make the system easier to use.

A user's computer should have a pointer (mouse), bit-mapped graphics display, and communications port (RS-232 or Ethernet) in order to perform cooperative user interface processing. Simple terminals (ANSI, VTxxx) are not candidates for CUIs.

For AdaNET, many users will have PCs, Macintoshes, or X-window Unix machines and therefore may take advantage of the benefits a CUI offers. Already, many commercial information services offer CUIs to PC or Mac users (e.g. U.S. Videotel, GENie, CompuServe, etc). These CUIs are either custom software (e.g. CompuServe's Navigator program) or MicroPhone [MIC] scripts (e.g. Dow Jones, GENie).

The remainder of this report presents the results of a HyperCard-based feasibility test for CUI technology, a brief survey of commercial tools that may be used to build CUIs for AdaNET, a discussion of protocol designs for data reduction, platform considerations, and recommendations for prototype development.

HyperCard Feasibility Prototype

To test the feasibility of a CUI for AdaNET, a simple prototype was built with HyperCard2, the serial XCMD toolkit, a Hayes modem, and an Opus bulletin board system (BBS) operated by Kurt Gutzmann (the BrainInAPan, FidoNet Node 1/106/260, 480-7422). The initial results were very encouraging.

The usual interaction of a user with the BBS is shown in a session log in Figure 1. This interface is a menu and command-line type, where the user enters the first letter of the command. The log shown below is for a session in novice mode; the commands are spelled out explicitly. A regular mode prompts only with the first letters, so the main menu prompt appears as: "MFGSABYCUEVO!:" A further reduction of data is provided by the expert mode, where no prompt is given.

¹, the benefit of this is a faster and more responsive user interface resulting in greater user satisfaction with the system.

BBS systems have been operated by hobbyists for over a decade now. During much of that time, only 300 or 1200 bps modems were available. Hence, BBS developers and operators have paid much attention to reducing the data transmission requirements.

The HyperCard prototype replaces the command string entries with button-presses. Pressing a button causes the Macintosh to emit the appropriate command string from the communications port. Pop-up menus (not shown, but demonstrable) are used for message area and file area selections as well. In this system, the host BBS has no knowledge of the type of user interface on the other end of the connection.

atdt9,4807422

CONNECT 2400

OPUS-CBCS v1.03a

Brain In A Pan BBS - Houston TX USA

What is your FIRST name: kurt;gutzmann;y;<password>

Hello Kurt, this is your 5115th call to the Brain.

It's 29 Apr 91, 9:38:01 CST. You have 21 minutes available.

Last caller was * Laurent Bossavit Alfortville, France 29 Apr 91 7:12:56

** Brain In A Pan **

March 1 1990

The Brain has bought its modem, at last.

Its hopes to be a 2400 BAUD BRAIN have finally come true.

This system will be unavailable in 21 minutes.

MAIN MENU: Type '?' by itself for help

M)essage section	F)ile section	G)oodbye (logoff)	S)tatistics
A)ns Questions	B)ulletin	Y)ell at sysop	C)hange setup
U)ser list/search	E)xtra Info	V)ersion	O)utside
!)Sysop Menu			

Select: f

The FILES Section.

File area # 3 ... PC Files

A)rea change	L)ocate	F)ile titles	T)ype (show)
G)oodbye	U)pload	D)ownload	S)tatistics
M)ain menu	R)aw dir	K)ill file	C)ontents
H)url	O)utside file mgt.		

Select: a

File Areas -----

Opus SYSOP Files

- 1 ... General
- 2 ... Macintosh Files
- 3 ... PC Files
- 5 ... MIDI Files
- 6 ... Soaring Files
- 7 ... Hypercard Files
- 8 ... sysop private
- 16 ... Files obtained by File Robot requests

File area: 2

Figure 1. BBS Session Log

File area # 2 ... Macintosh Files

A)rea change	L)ocate	F)ile titles	T)ype (show)
G)oodbye	U)pload	D)ownload	S)tatistics
M)ain menu	R)aw dir	K)ill file	C)ontents
H)url	O)utside file mgt.		

Select: mm

The MESSAGE Section

There are 51 messages in this area. The highest is #51
The last message you read was 21

Msg.area 1 ... 'Nothing specific'

Type '?' by itself for help

A)rea change	N)ext (read msg)	P)rior msg	E)nter message
R)eply	=)read non-stop	-)read original	+)read reply
L)ist (brief)	S)can	I)nquire	M)AIN MENU
G)oodbye (logoff)	K)ill message	H)url (move)	F)orward (copy)
O)utside	U)pload	X)port to disk	

Select: g

Disconnect [Y,n,?=help]?

Leave a note to Kurt Gutzmann [y,N,?=help]?

Logging your call...

Bye Kurt Gutzmann, thanks for calling. Please hang up now.

Figure 1. BBS Session Log - continued

Text capture functions are easily implemented (the FileList window displays various captured data). The serial port XCMDs were not always reliable. Figures 2,3,4, and 5 show the CUI implemented in HyperCard II.

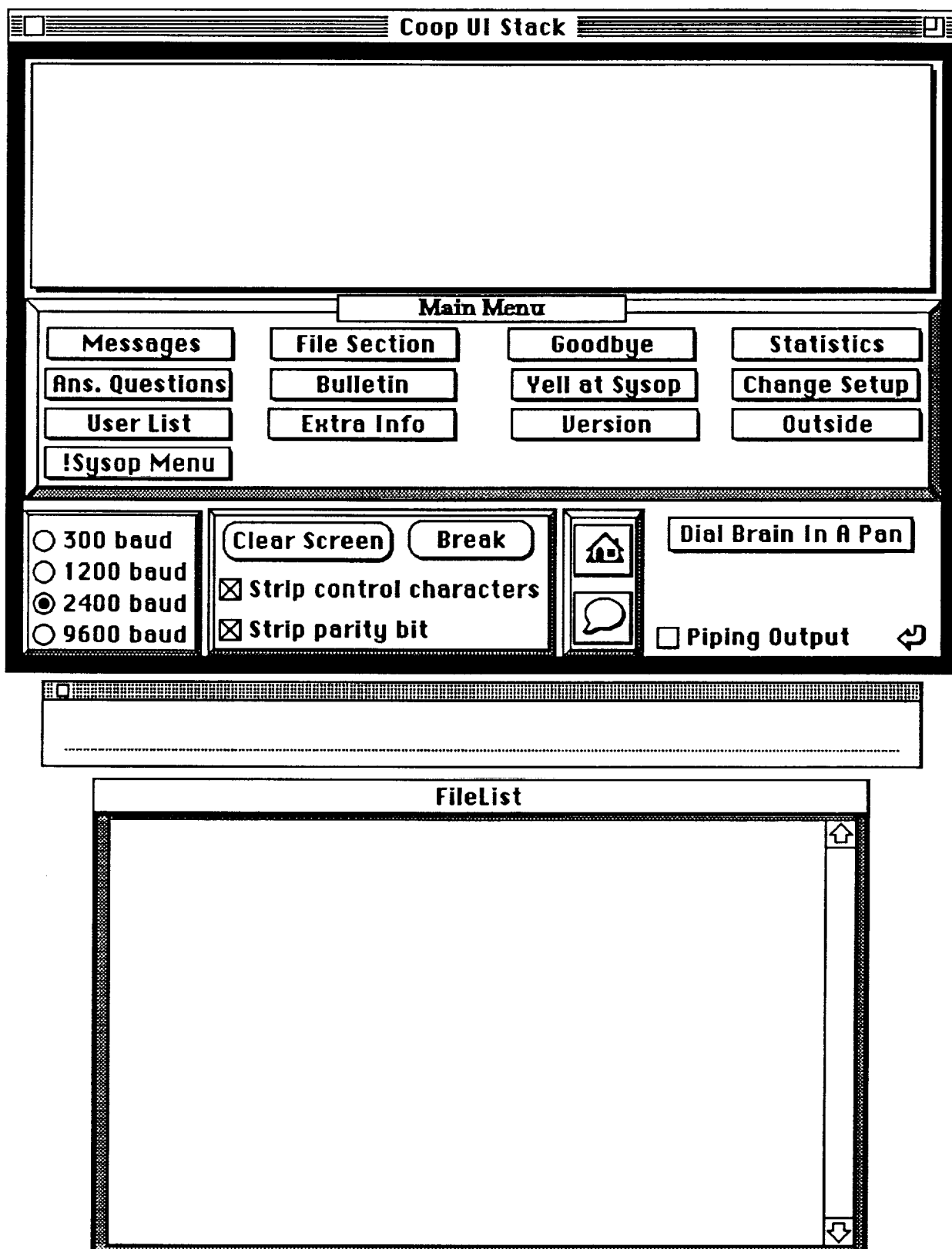


Figure 2. CUI for BBS, Initial Screen

Coop UI Stack

File Section

Area Change

Locate

File Titles

Type

Goodbye

Upload

Download

Statistics

Main Menu

Raw Directory

Kill File

Contents

Hurl

Outside Files

☐ 300 baud
☐ 1200 baud
☒ 2400 baud
☐ 9600 baud

Clear Screen

Break

☒ Strip control characters
☒ Strip parity bit

☐ Piping Output

↶

Figure 4. File Area Screen

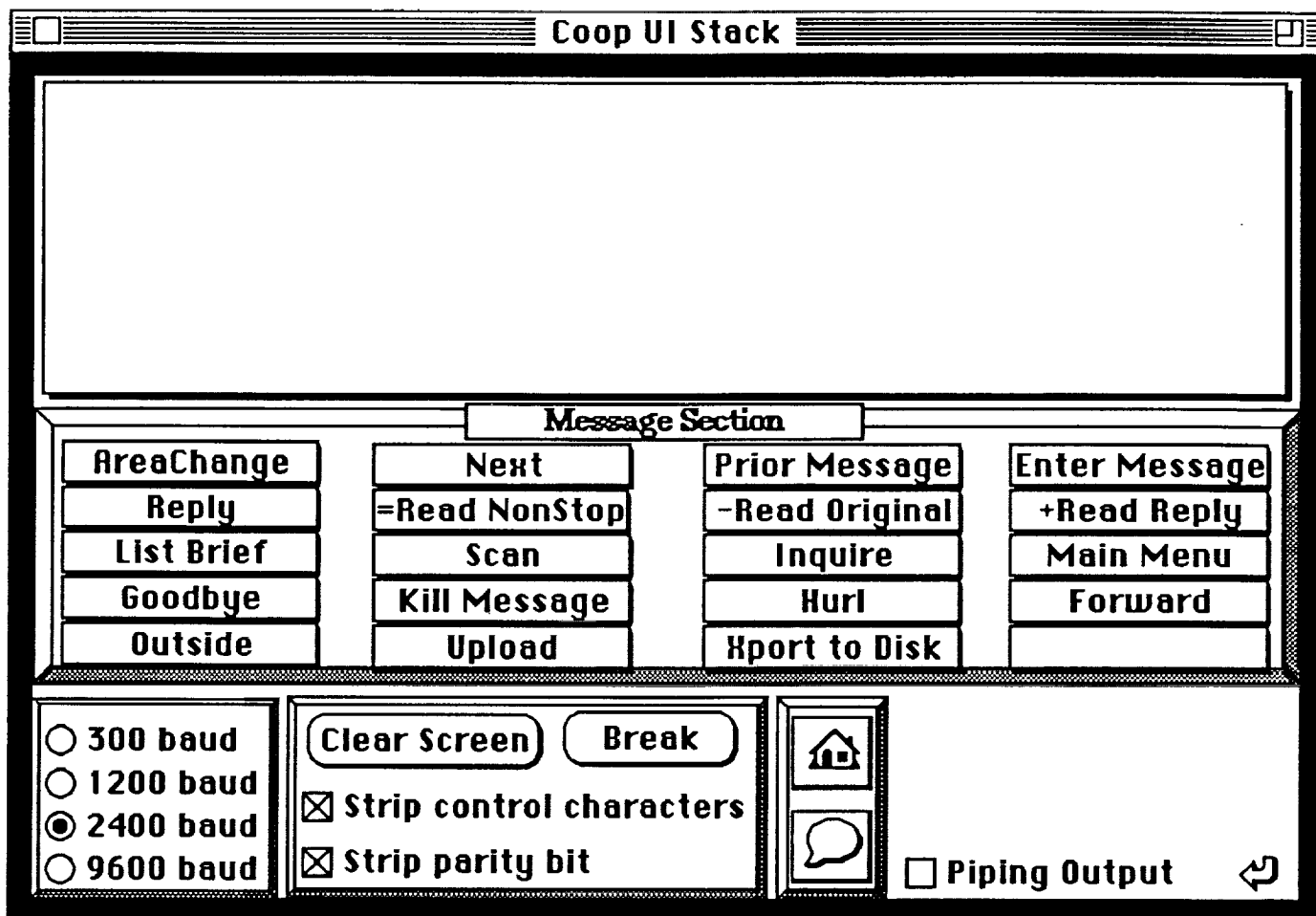


Figure 5. Message Area Screen

It is evident that the graphical interface is simpler and friendlier than the single-letter command line interface. Some systems, such as the current ASV2 require complex keystroke combinations which further reduce system usability.

The various features of this prototype are available for demonstration at any time.

Commercial Interface Builders

Many products are available to assist in the development of CUIs. Many more screen management systems are available than are integrated screen and communications systems. The first category includes products such as:

- IntegrAda Screen Tool (PC)
- JIAC Application Manager (JAM, for PC)
- ScreenMaker (PC)
- Prototyper 3.0 (Mac)
- Dialog Builder (Mac)

- HyperCard & ToolBook (Mac and PC)

MicroPhone is one of the few products in the second category (screen and communication systems).

MicroPhone II

MicroPhone II is an outstanding commercial off the shelf product intended for the development of CUIs. Many commercial information services provide MicroPhone scripts for their systems (BIX, CompuServe, Delphi, Dow Jones, EasyLink, GENie, MCI Mail, and The Well).

MicroPhone is available for both PC and Macintosh platforms. We have communicated with Software Ventures, the developer, and are now on their beta-test site list. Consequently, we shall receive the forthcoming releases of MicroPhone for Windows 3.0, which uses dynamic data exchange (DDE) functions that allow MicroPhone to operate as a communications engine for other applications. A similar release for Macintosh System 7.0 using interapplication communications (IACs) is planned for release in summer of 1991. These systems will enable straightforward construction of CUIs since no development effort need be diluted into low level communications programming, focusing instead on the user interface (which may be built using the screen-only tools listed on the previous page).

An example screen for CompuServe is shown below in Figure 6. The buttons in the lower region of the screen change as the user navigates through different areas of the CompuServe system. Almost all navigation is performed by pressing buttons.

This type of application is easily customized for any online system. A drawback to this approach is that no data reduction is obtained, since the MicroPhone scripts are keyed to the usual character sequence emitted by the host. However, user friendliness is greatly improved.

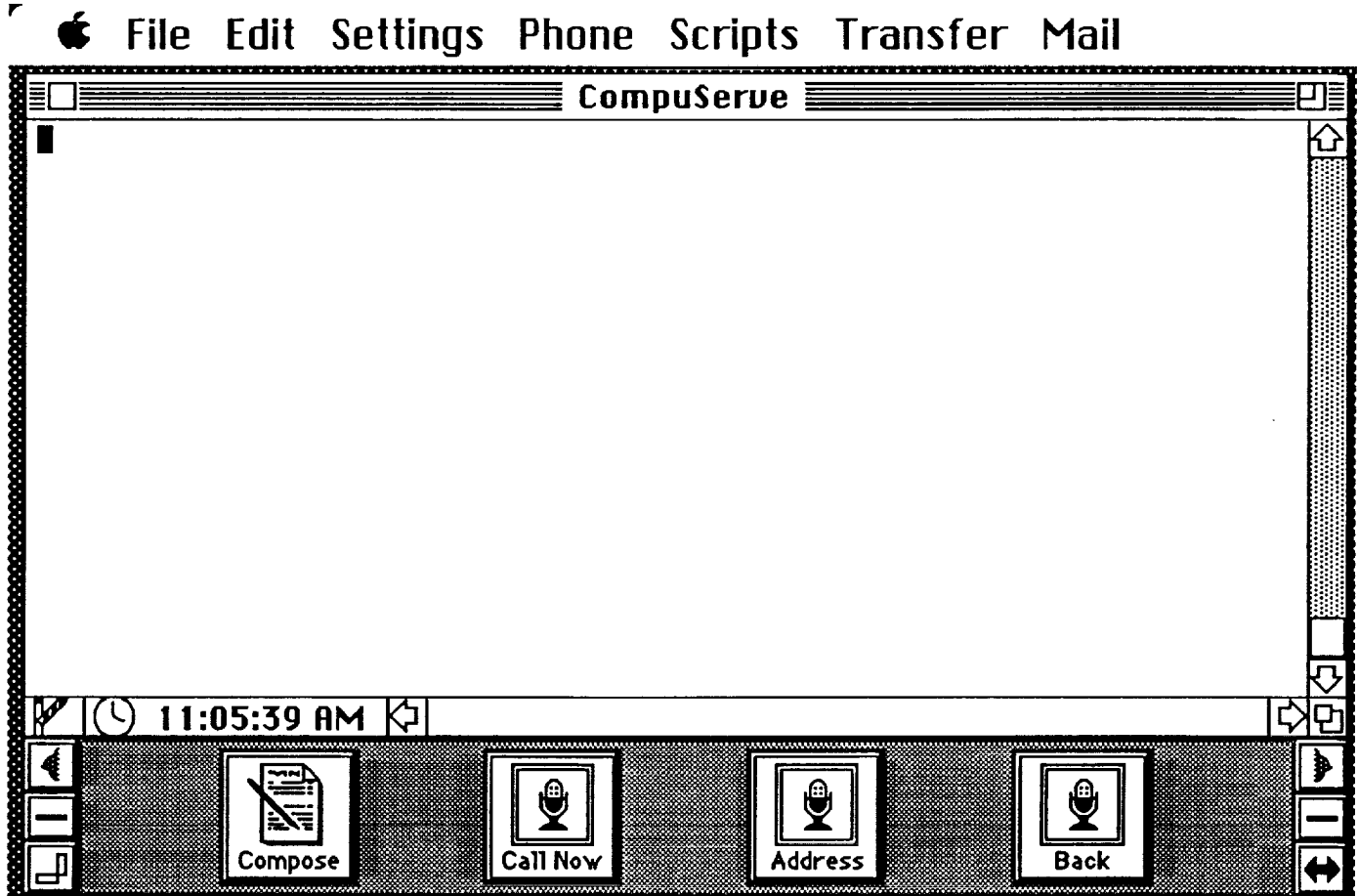


Figure 6. MicroPhone II CompuServe Screen (initial)

CUI Communications Protocol

For production implementation, it is not sufficient to simply shield the user from the native host data stream. An approach like that of X-windows, with a compact tokenized UI control protocol is best. This minimizes the amount data transmitted (a critical concern for online dial-up systems like AdaNET).

The basic elements of a CUI protocol are:

- host and user system state machine definitions
- packet type definitions (control and data) and protocol parameters
- user interface element definitions
- mechanisms for synchronization of state machines on the host and the user computer
- verification of state changes and packet integrity (error checking and correcting)

All of the complex graphics are stored or constructed locally by the user's computer, and are presented in response to a control packet received from the host or transmitted to the host and acknowledged by it.

A CUI protocol should be developed first and be sufficiently generic so that instantiations of it for various CUI systems are simple and easy to implement.

Platform Considerations

It is often tempting to use the lowest common hardware denominator for user computers in order obtain the largest number of qualifying platforms. In this case, the lowest common denominator is probably a PC with a character-only display without mouse. This is slightly better than a dumb terminal. Accordingly, it is difficult to obtain any benefits of the CUI with a lowest common denominator platform.

The other extreme in hardware would be to require a workstation-caliber machine, in which case very few users (relative to the lowest common denominator) have access to CUI benefits. This is also unreasonable.

We should strive to take advantage of the strong points of the common personal computer and workstation hardware today. Many PCs now have a bit-mapped graphic display and mouse (capable of executing Windows 3.0). All Macintoshes are ideally suited to CUI processing. And most engineering workstations today support bit-mapped displays and mice. The basic CUI for AdaNET should be developed using X-windows (available on Unix workstations, PCs, and Macintosh), with modification for compact data transmission and local storage and generation of bit-map displays. Excellent performance over 1200 or 2400 bps lines should be attainable this way.

Conclusion

AdaNET ASV3 and ASV4 could reap significant benefits from a CUI system element. The most common complaint users have of online systems is that they are slow and hard to use. CUI technology directly addresses these problems. When users have a rewarding experience with a new system, they tell others about; for AdaNET, this could be an extremely effective way of generating a larger user base. The CUI elements should be developed to take advantage of the features unique to each of the PC, Macintosh, and Unix workstation platforms. Simultaneously, the CUI should share as much code as possible among the three platforms. X-windows enables significant amounts of code-sharing among these. Communications functions should be purchased off the shelf where possible, and integrated by IPC, DDE, and IAC with the user interface functions. The next step in this effort should be the design of a general purpose CUI protocol and prototype implementation of it. With a robust CUI protocol, development of CUI elements for various platforms is straightforward.

References

- [MIC] Software Ventures Corporation, MicroPhone II Reference Manual, 1990.
- [X11] The User Interface Component of the Applications Portability Profile (MIT X Version 11), FIPS number 158;